RSAC 2025 Conference

Many Voices.
One Community.

SESSION ID: SBV-W08

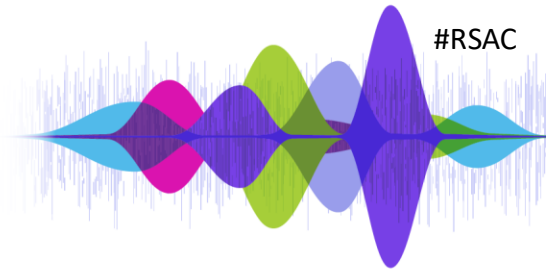
When Regulation Backfires: How a Vulnerable Plugin Led to an XSS Pandemic

Ori Ron

Senior Security Researcher
Checkmarx
https://www.linkedin.com/in/ori-ron64/

Eilon Cohen

Security Analyst
Checkmarx
https://www.linkedin.com/in/eilon-cohen/



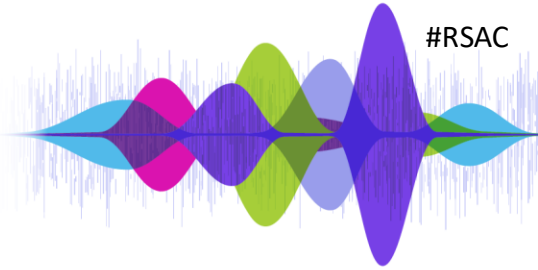
Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Agenda



- The Background Story
- Regulations in Action
- Scaling Up to a Pandemic
- Supply Chain Security
- Summary & Takeaways

RSAC 2025 Conference

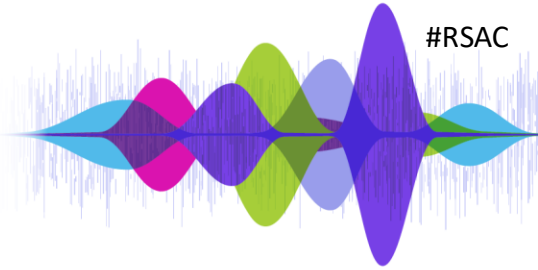
The Background Story

How This All Started

Many Voices.
One Community.

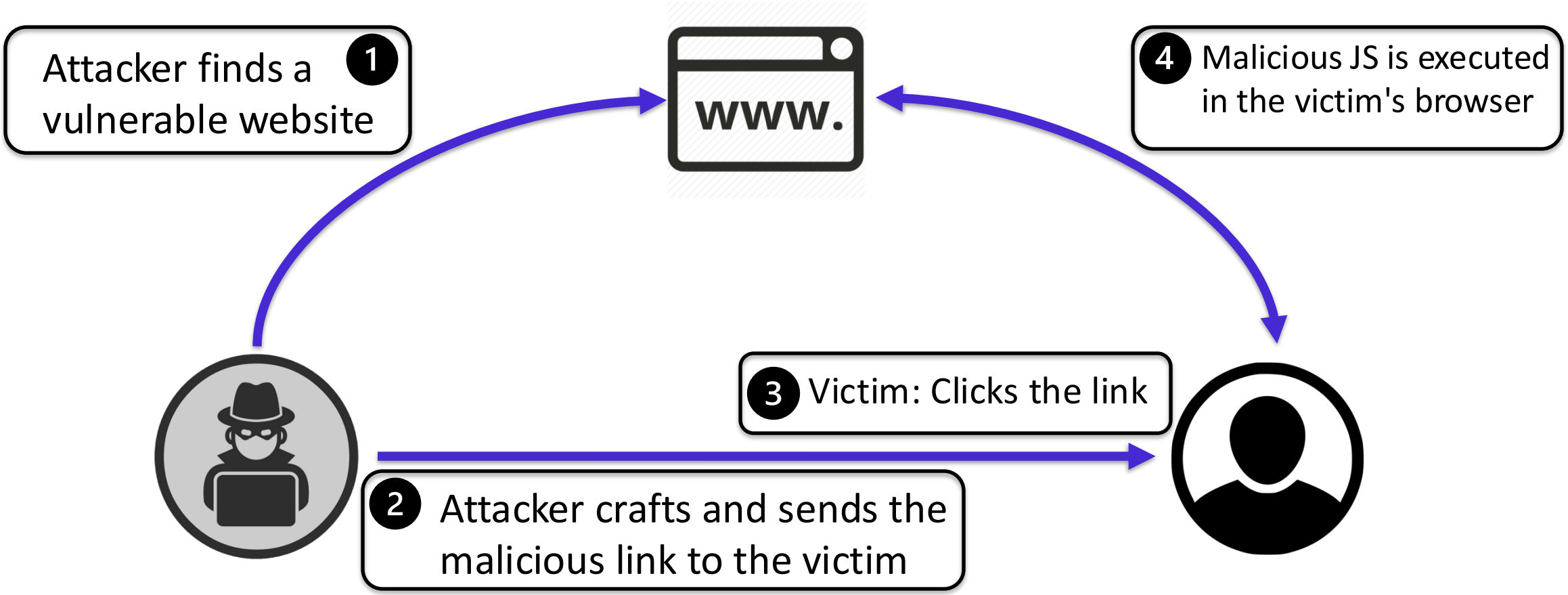
The WAF Challenge

XSS in a Nutshell



- Script injection to benign websites
- Different Types of XSS
  - Stored
  - Reflected
  - DOM

XSS – Typical Attack Flow



XSS in a Nutshell

- **Impact**
  - Gaining access to sensitive info related to the victim
  - Changing the page presented to the user
  - Taking actions on behalf of the victim
  - Chaining XSS with other vulnerabilities
- **Mitigations**
  - HTML Encoding
  - Input Validation
  - CSP Header

The Vulnerable Code

```
let img = "<img id='plugin-logo' src='https://plugin-domain.it/logo.png?uri=";

img += window.location.href;

img += "' />";

document.getElementById('navbar').innerHTML = img
```

The Vulnerable Code

```
let img = "<img id='plugin-logo' src='https://plugin-domain.it/logo.png?uri=";

img += window.location.href;

img += "' />";

document.getElementById('navbar').innerHTML = img
```

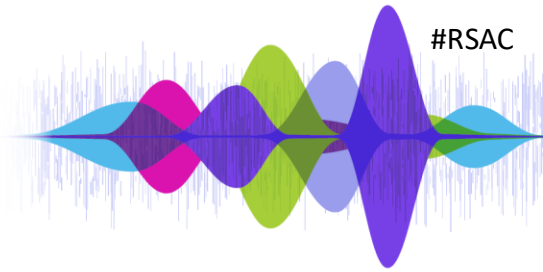
The Vulnerable Code

```
let img = "<img id='plugin-logo' src='https://plugin-domain.it/logo.png?uri=";

img += window.location.href;

img += "' />";

document.getElementById('navbar').innerHTML = img
```

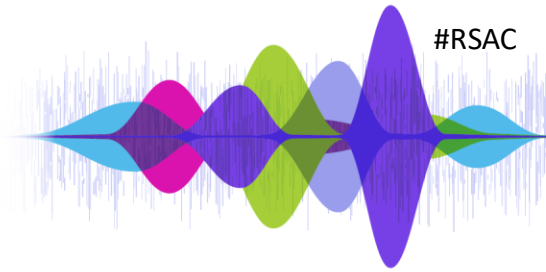
The Vulnerable Code

```
let img = "<img id='plugin-logo' src='https://plugin-domain.it/logo.png?uri=";

img += window.location.href;

img += "' />";

document.getElementById('navbar').innerHTML = img
```

HTML Attributes - Events

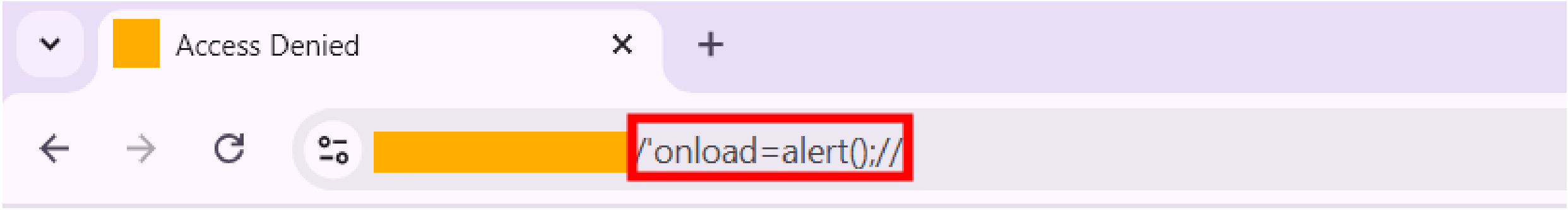


- Injecting HTML attributes to the element
- Useful events:
  - onload
  - onerror



```
'onload=alert();//
```

Executing the Payload



Access Denied

You don't have permission to access "http://[redacted]/%27onload=alert();/" on this server.

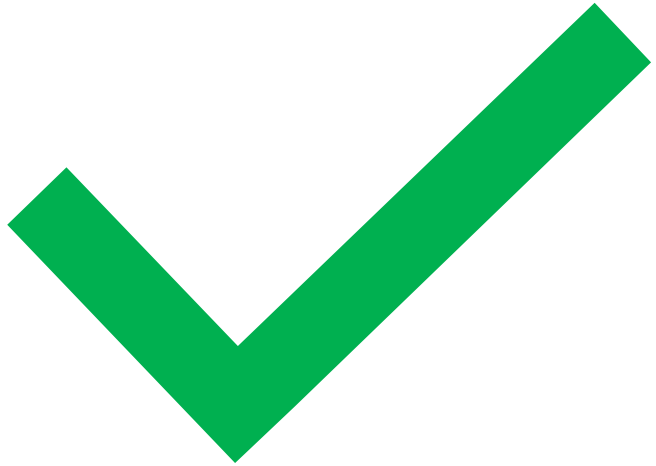
The First Bypass

Blocked

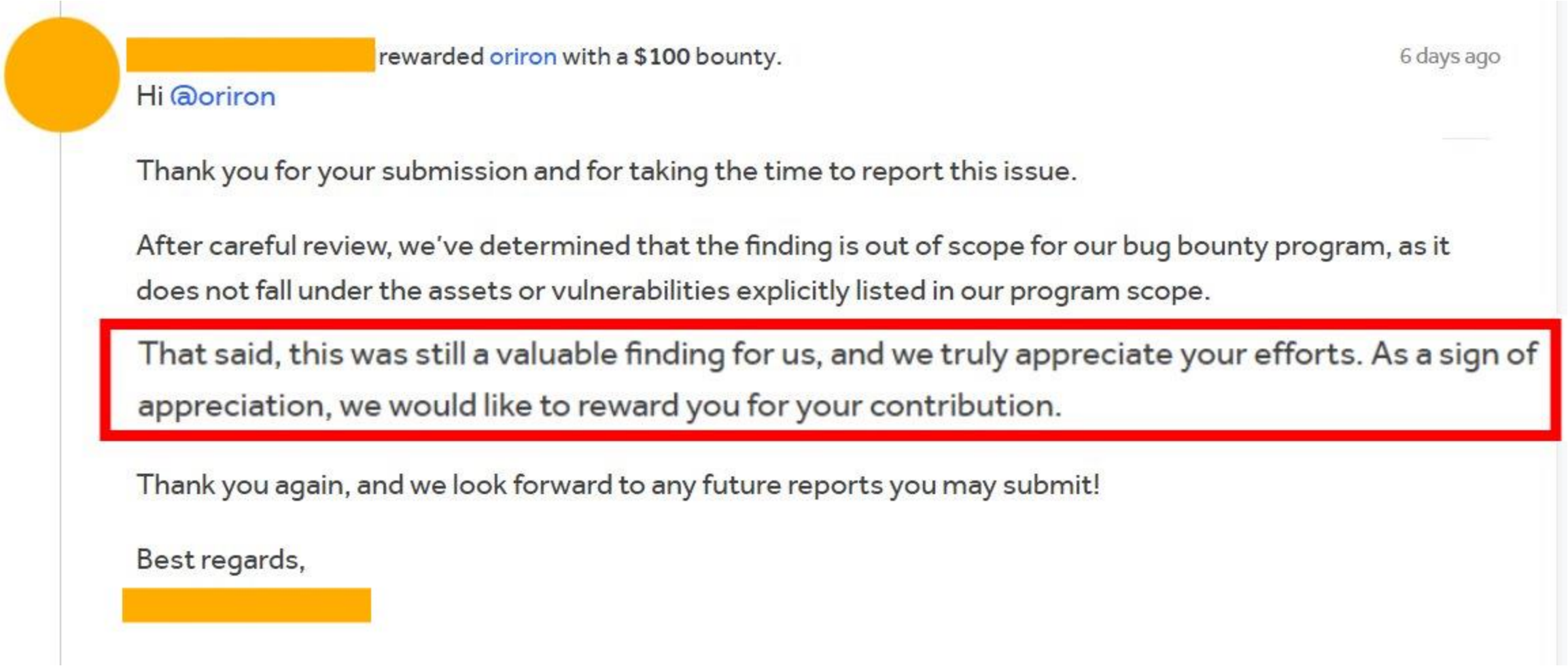
- ’onload=alert();//

Bypass

- ’onload=/**/alert();//



WAF Bypassed!



[redacted] rewarded oriron with a $100 bounty. 6 days ago

Hi @oriron

Thank you for your submission and for taking the time to report this issue.

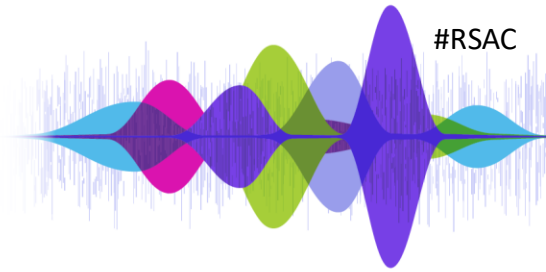
After careful review, we've determined that the finding is out of scope for our bug bounty program, as it does not fall under the assets or vulnerabilities explicitly listed in our program scope.

That said, this was still a valuable finding for us, and we truly appreciate your efforts. As a sign of appreciation, we would like to reward you for your contribution.

Thank you again, and we look forward to any future reports you may submit!

Best regards,

[redacted]



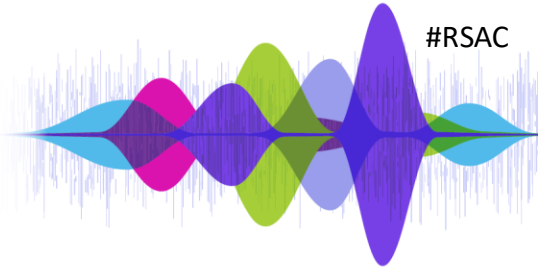
The Root Cause:

External Plugin's Vulnerability

RSAC 2025 Conference

Regulations in Action

The Accessibility Regulation - Why

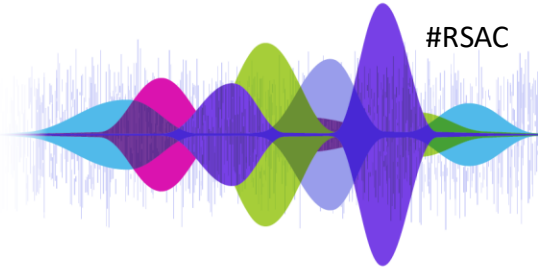


- Make the digital world inclusive, for example:
  - **Alternative text for images** helps screen-readers users
  - **Better color contrast** helps people with low vision
- Not only compliance but also **better usability for everyone!**

Accessibility Around the World

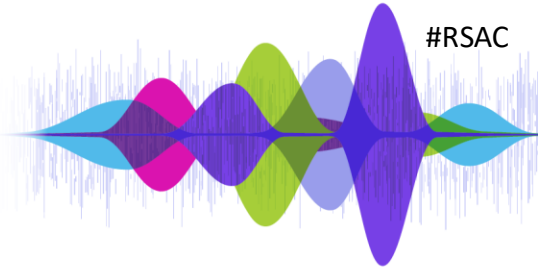
- WCAG 2.2
  - 3 Different Conformance Levels
- WCAG 3
  - Currently an incomplete draft
- Other Regulations
  - USA – ADA
  - Europe – EEA and WAD

Accessibility – The Challenges



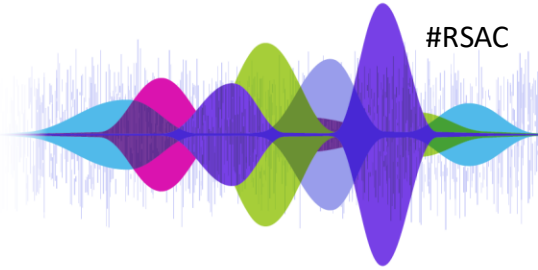
- Varying Regulations
- Evolving Standards
- Limited Internal Expertise
- Fear of litigation

Accessibility – The Consequences



- A variety of third-party plugins emerged and were rapidly adopted.
- Compliance prioritized over security

It’s Not Just About Accessibility

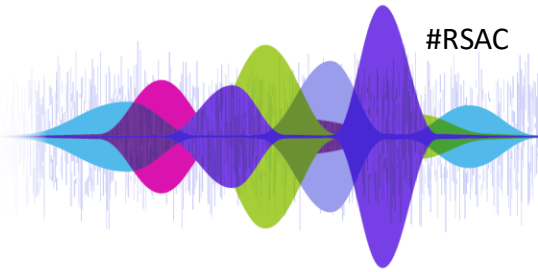


- Privacy & Cookie Consent - GDPR
- Age Verification Plugins – COPPA, Alcohol laws
- Health Data Protection - HIPAA

RSAC 2025 Conference

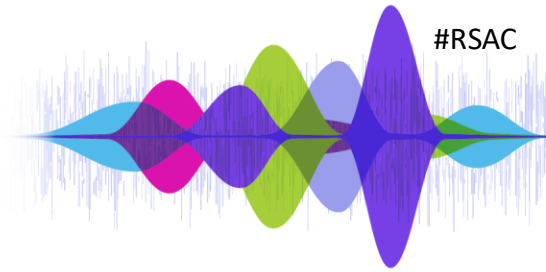
From A Single XSS to Pandemic

Setting Up the Stage



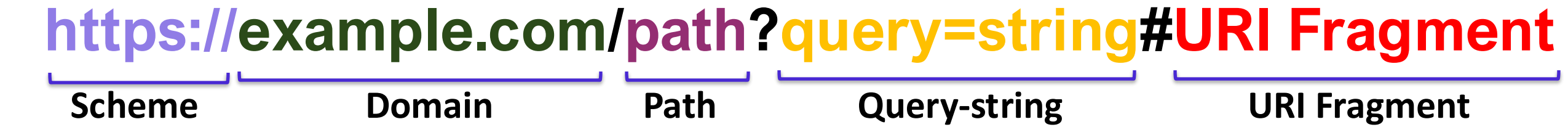
The Plugin Populairty

- The regulation:
  - Requires many organizations to be compliant
- The plugin:
  - Has been provided free of charge (for years)
  - Has simple installation process

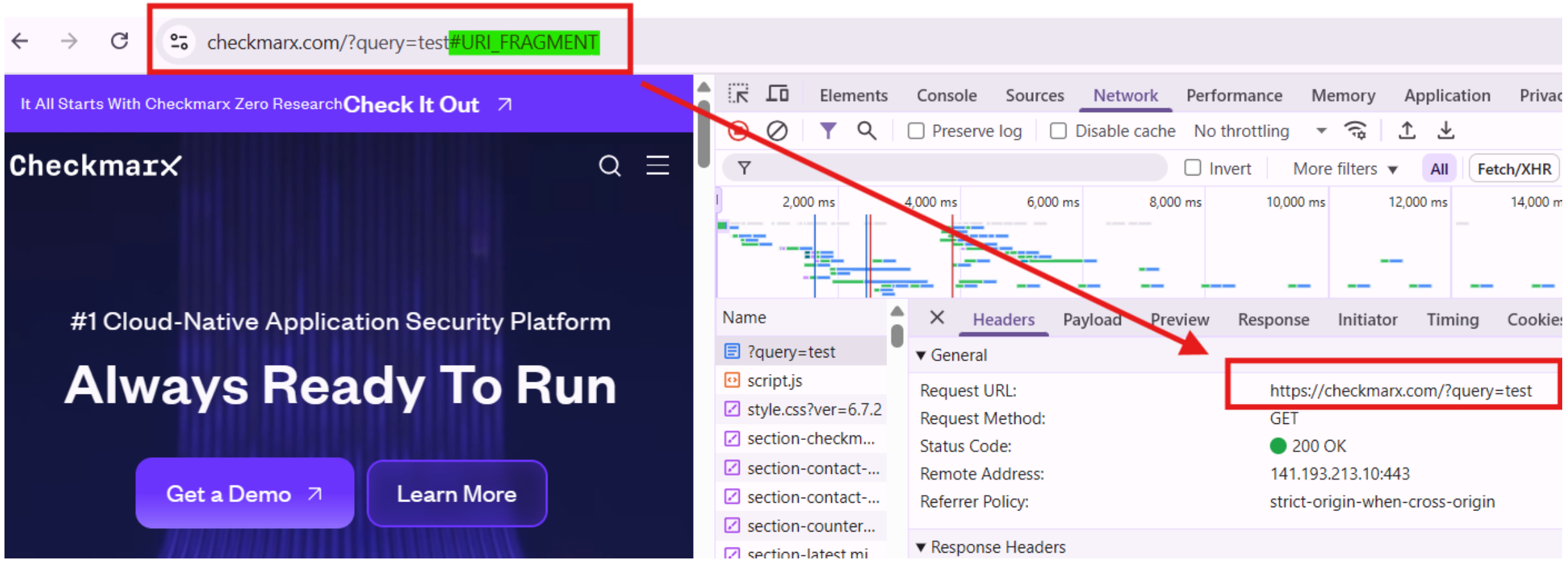


The plugin must be
POPULAR

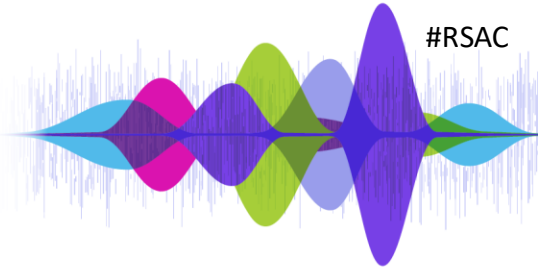
The URI Structure



The URI Fragment



The URI Fragment - Undetectable Payload



- Evading any WAF
- Evading sever-side components, for example:
  - Protection mechanisms
  - Logging

RSAC 2025 Conference

From A Single XSS to Pandemic

The Outbreak

Target Discovery Process

Targets Collection

Google Dorking

Source Code Search Engines

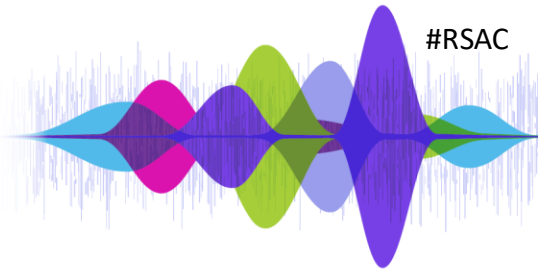
Exploitability Verification

An automated verification

Affected Targets Mapping

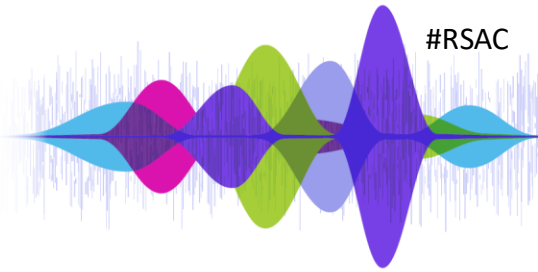
Manual analysis to assess the risk & impact

Exploitability Verification at Scale



- **What**
  - Automated way to verify XSS in over 1,000 websites
- **How**
  - Option #1: Writing our own program to verify the targets
  - Option #2: Using an open-source tool

Exploitability Verification at Scale

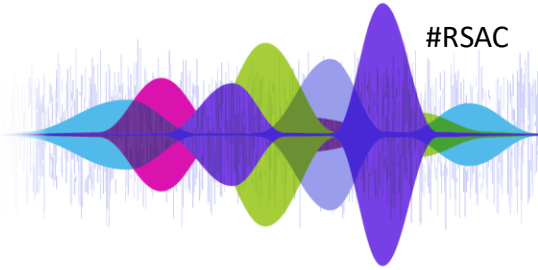


- **Requirements**
  - Browser capabilities
  - JS execution detection
  - Eliminating FPs/FNs
  - Standardized way for testing websites

Nuclei it is!

- We decided to use Nuclei Templates
- Payload is in URI fragment
- Exploitation is confirmed when **`alert()`** dialog is popped
- Benefits:
  - General standardized YAML rule
  - Scales easily

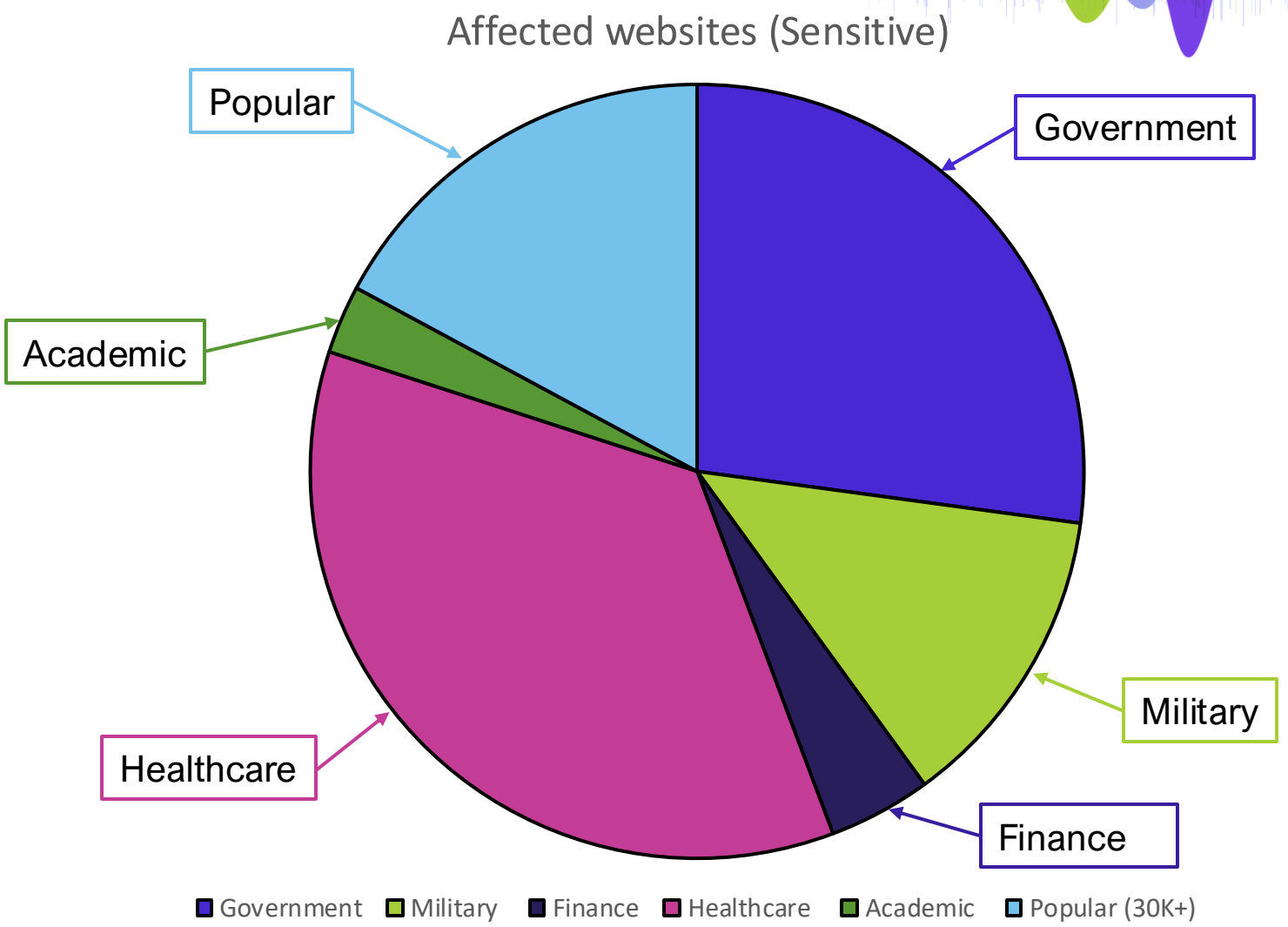
Assessments & Disclosures



- Assessing Website Sensitivity
  - Sector (Military, governmental, etc.)
  - Popularity (SEO rank, traffic, etc.)
  - Functionality
- Disclosure
  - First disclosure to CERTs
  - Individual reports for sensitive organization

Stats

- 1000+ Affected websites
  - 25 Healthcare
  - 19 Government
  - 12 Popular
  - 9 Military
  - 3 Finance
  - 2 Academic



RSAC | 2025 Conference

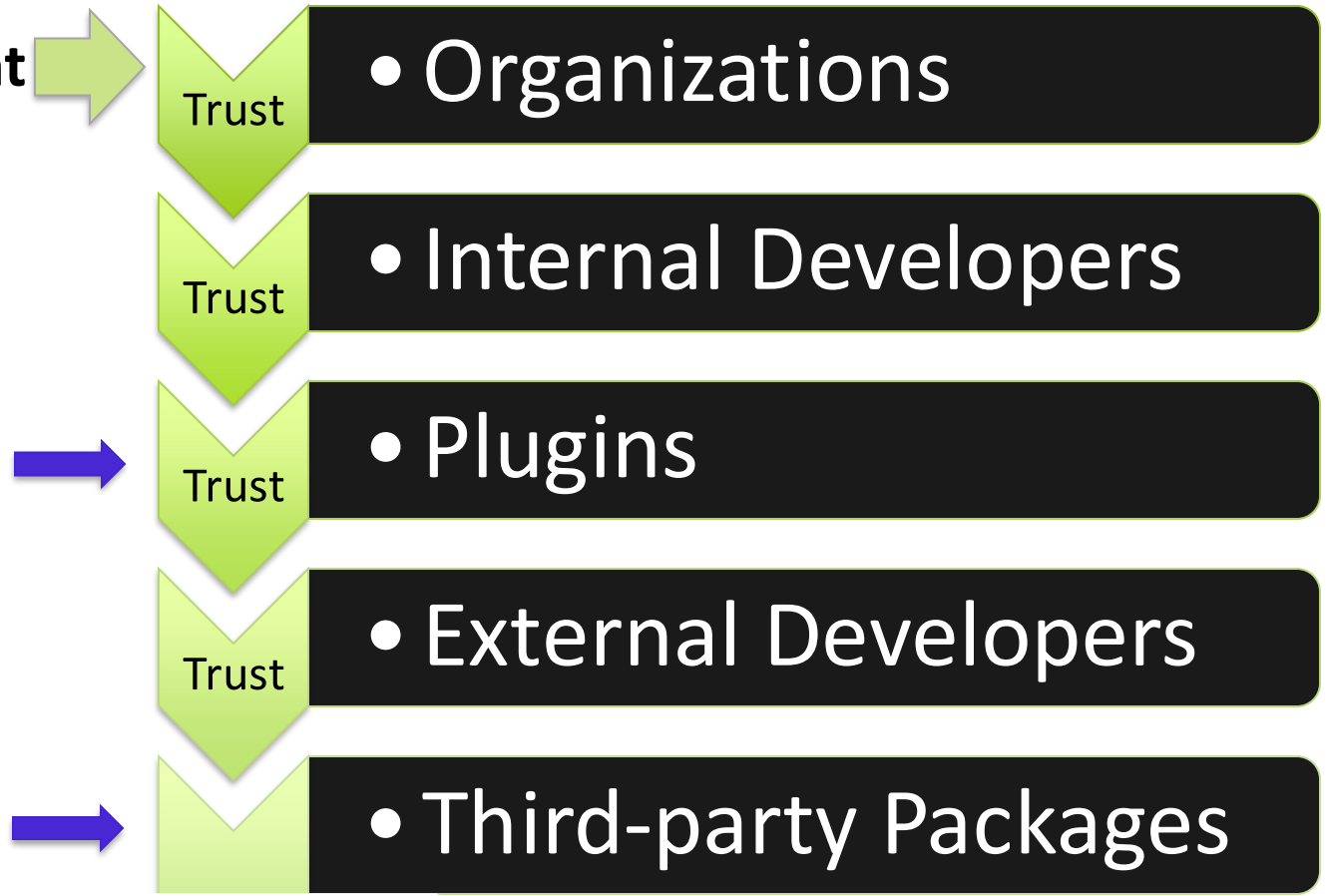
Supply Chain Security

The Devil is in the Plugins

Many Voices.
One Community.

Regulation Chain of Trust*

The Regulator Initiates a
Compliance Requirement



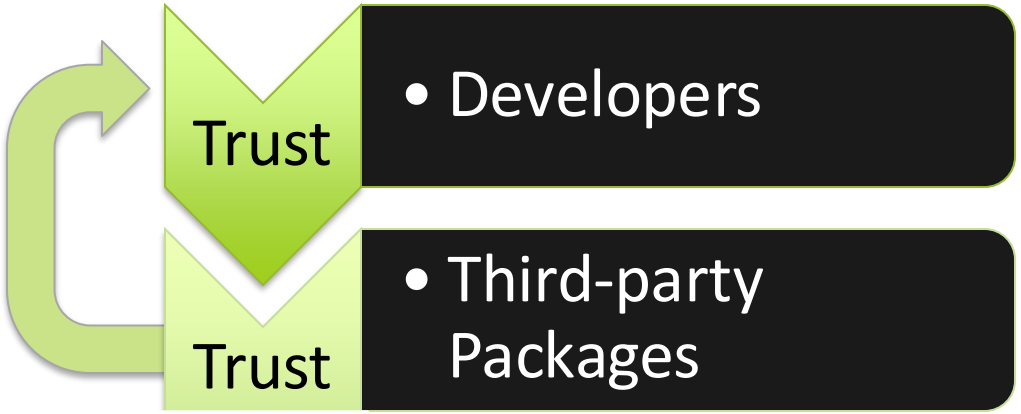
NOTE: Trust does NOT end here!

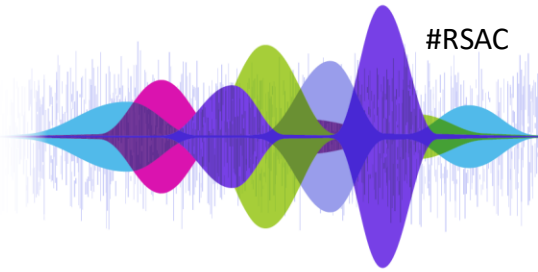
In fact, trust does NOT even end here

*** Not to be confused with the "Certificate Chain of Trust"**

The Never-Ending Trust

The truth is that the trust between developers to 3rd-party packages is a never-ending cycle!

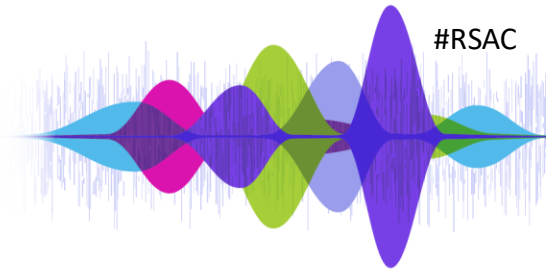




Supply Chain Risks

- The plugin may have security vulnerabilities:
  - Flaws in the plugin's source code
  - Vulnerabilities/malicious code in its dependencies
- The plugin itself may be malicious, for example:
  - It contains harmful code
  - Malicious code could be introduced in future updates

Supply Chain Risks



- Malicious actors may compromise the maintainer’s assets:
  - Through social engineering attacks
  - Exploiting infrastructure vulnerabilities

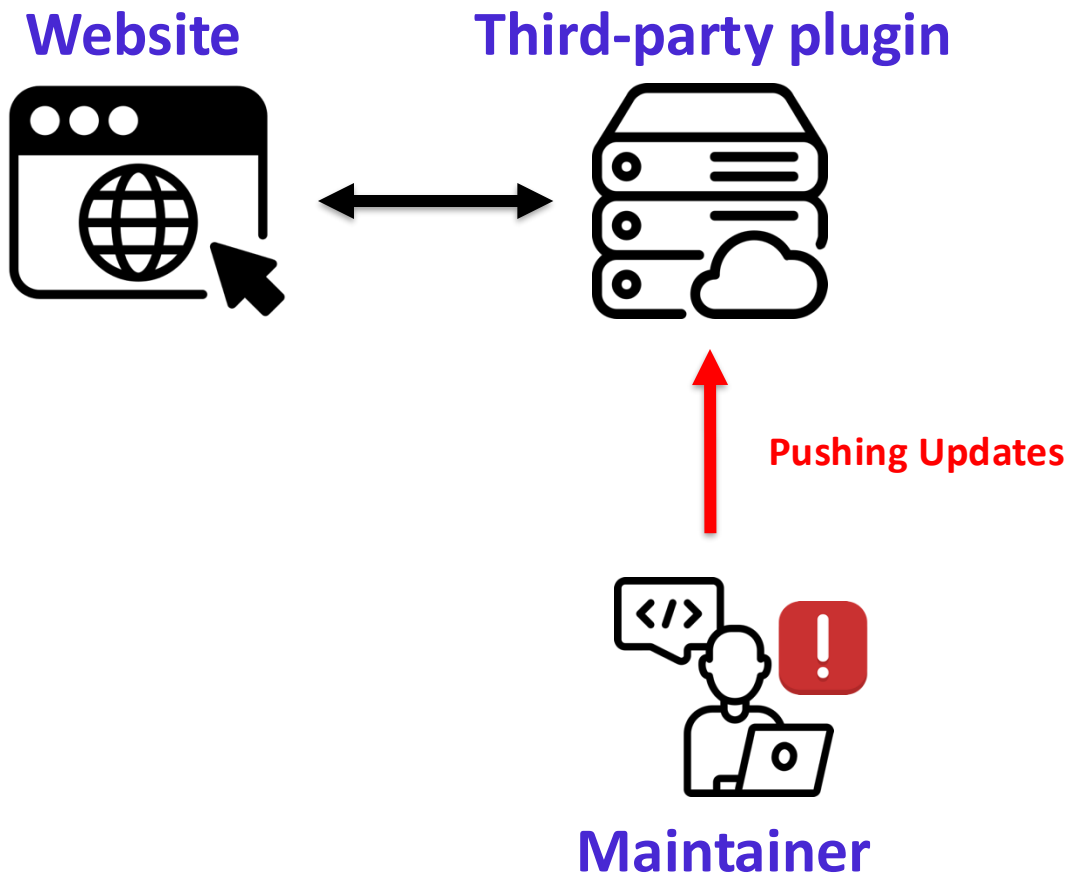
Hotlinking - Simplicity & The Trap

- Easy Integration, only one line of code
- Provides Instant Functionality
- Maintained outside of the organization

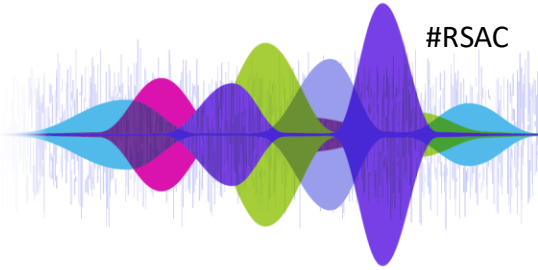
```
1 <head>
2   ...
3   <script src='//3rd-party.com/widget.js'></script>
4   ...
5 </head>
```

Hotlinking - The Trap: What You Risk

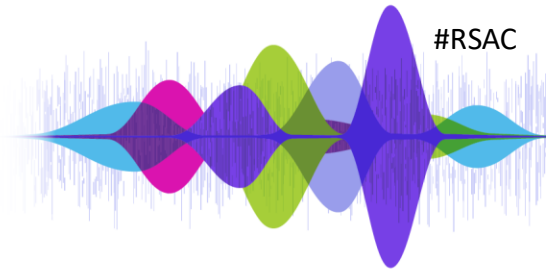
- Loss of Control / Versioning
- Visibility Gap / Blind Spot
- Inherited Risk



Reducing Supply Chain Risks

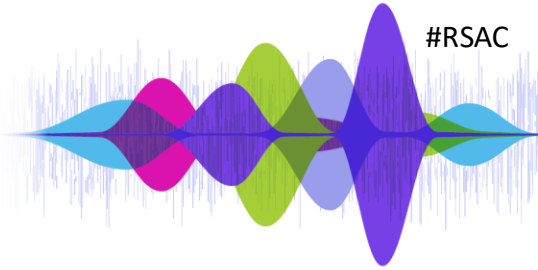


- Vet all your 3rd-party dependencies & vendors
  - Can you REALLY trust them?
- Review security advisories & past drama
  - Are there vulnerabilities left without a proper fix?



You can't be 100% safe!

Supply Chain Risks – In Practice



Key issues we identified:

- ☑ Vulnerable dependencies
- ☑ Malicious code introduced after plugin gained popularity
- ☑ Flaws in the plugin's source code itself

RSAC 2025 Conference

Takeaways

Takeaways

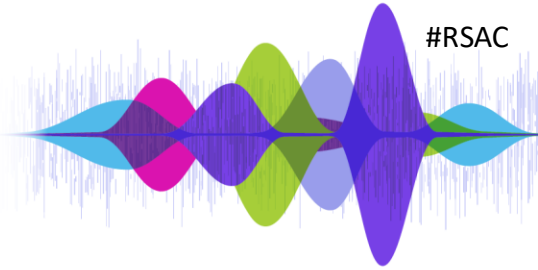
Defenders

- Don't blindly trust code you didn't write
- Keep tracking your 3rd-party packages and plugins

Researchers / PTs

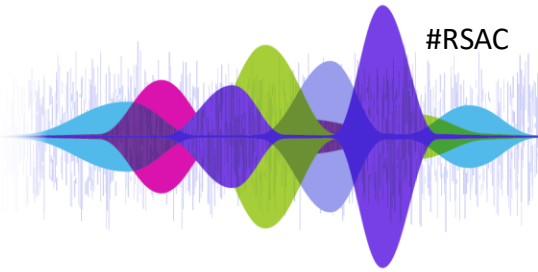
- Always look at the root cause of your finding
- Take your findings seriously

Apply What You Have Learned Today



- **Step #1: Visibility**
  - Identify which third-party plugins/libraries your org uses
  - Validate who is behind the plugin's source
  - Review the plugin security track record, and permissions

Apply What You Have Learned Today



- **Step #2: Assessment & Analysis**
  - Review and assess the dependencies behavior
  - Analyze whether it does suspicious activity or has vulnerabilities
- **Step #3: Respond**
  - Establish a process to vet and monitor third-party scripts
  - Educate developers and decision-makers on supply chain risks

RSAC 2025 Conference

Many Voices.
One Community.

Thank You

For more information:

ori.ron@checkmarx.com

eilon.cohen@checkmarx.com